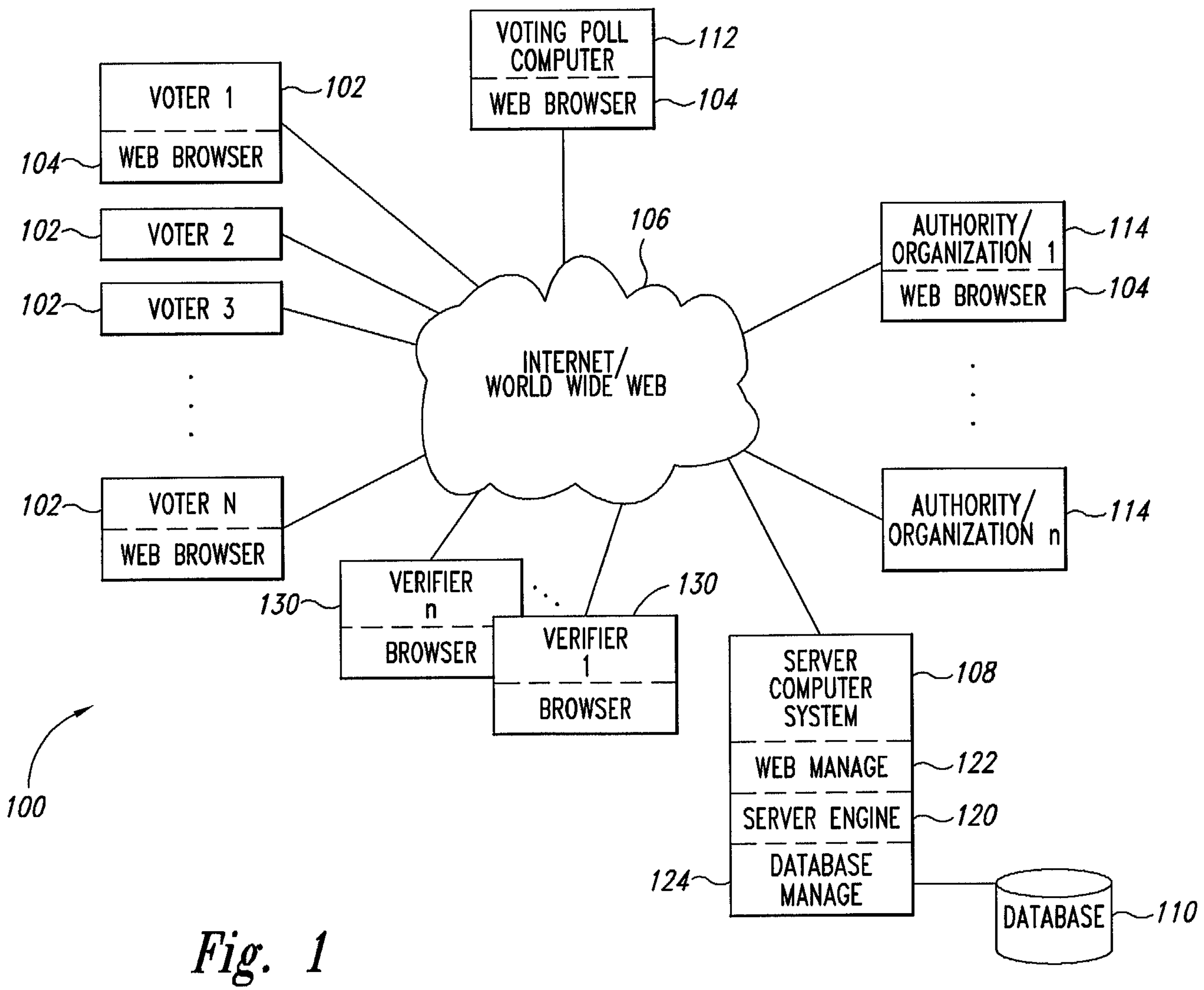

VOTER 1
WEB BROWSER
102
104
VOTER 2
102
VOTER 3
102
VOTER N
WEB BROWSER
102
VOTING POLL COMPUTER
WEB BROWSER
112
104
106
INTERNET/ WORLD WIDE WEB
AUTHORITY/ ORGANIZATION 1
WEB BROWSER
114
104
AUTHORITY/ ORGANIZATION n
114
VERIFIER n
BROWSER
130
VERIFIER 1
BROWSER
130
SERVER COMPUTER SYSTEM
108
WEB MANAGE
122
SERVER ENGINE
120
DATABASE MANAGE
124
DATABASE
110
100

Fig. 1

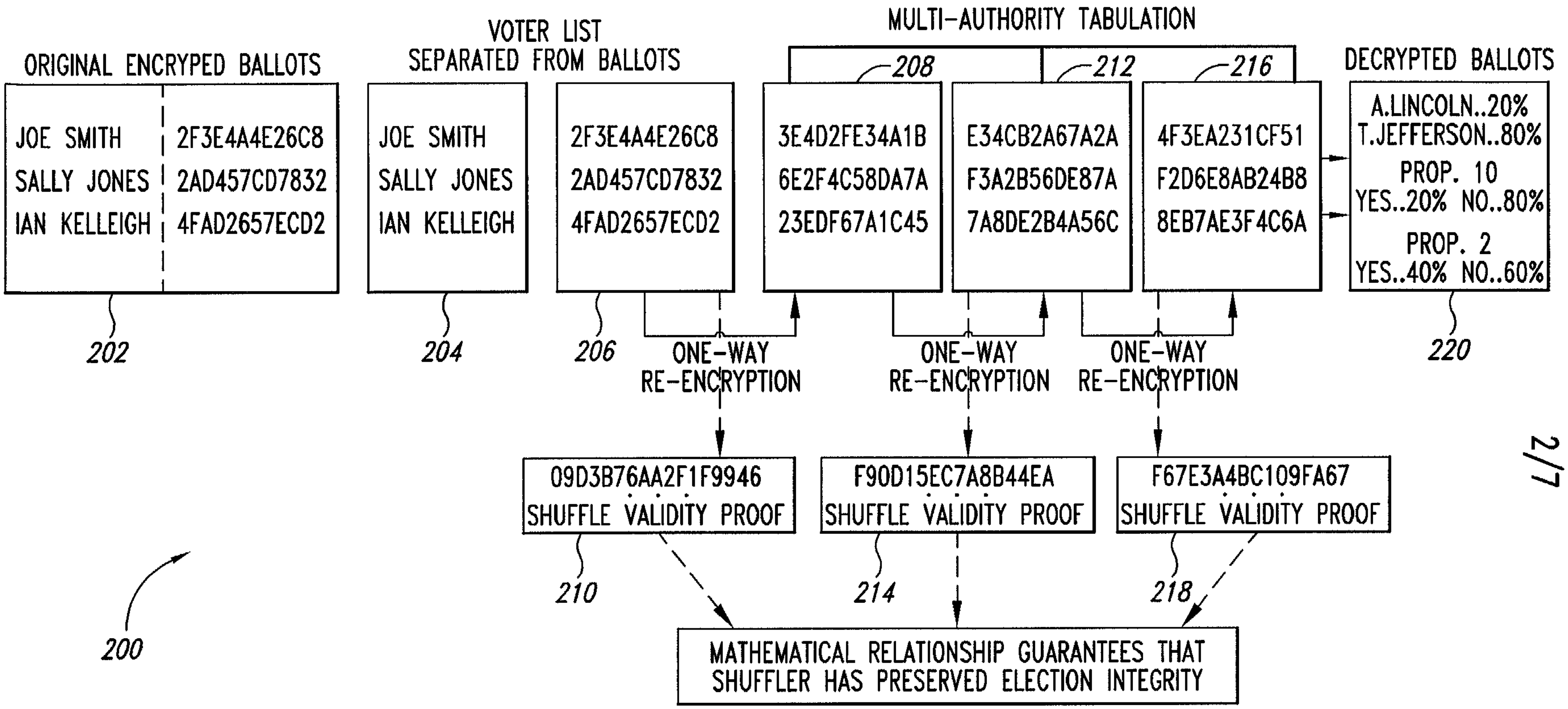

ORIGINAL ENCRYPED BALLOTS
JOE SMITH
SALLY JONES
IAN KELLEIGH
2F3E4A4E26C8
2AD457CD7832
4FAD2657ECD2
202
VOTER LIST SEPARATED FROM BALLOTS
JOE SMITH
SALLY JONES
IAN KELLEIGH
204
2F3E4A4E26C8
2AD457CD7832
4FAD2657ECD2
206
MULTI-AUTHORITY TABULATION
208
3E4D2FE34A1B
6E2F4C58DA7A
23EDF67A1C45
212
E34CB2A67A2A
F3A2B56DE87A
7A8DE2B4A56C
216
4F3EA231CF51
F2D6E8AB24B8
8EB7AE3F4C6A
DECRYPTED BALLOTS
A.LINCOLN..20%
T.JEFFERSON..80%
PROP. 10
YES..20% NO..80%
PROP. 2
YES..40% NO..60%
220
ONE-WAY RE-ENCRYPTION
ONE-WAY RE-ENCRYPTION
ONE-WAY RE-ENCRYPTION
09D3B76AA2F1F9946
SHUFFLE VALIDITY PROOF
210
F90D15EC7A8B44EA
SHUFFLE VALIDITY PROOF
214
F67E3A4BC109FA67
SHUFFLE VALIDITY PROOF
218
MATHEMATICAL RELATIONSHIP GUARANTEES THAT SHUFFLER HAS PRESERVED ELECTION INTEGRITY
200

Fig. 2

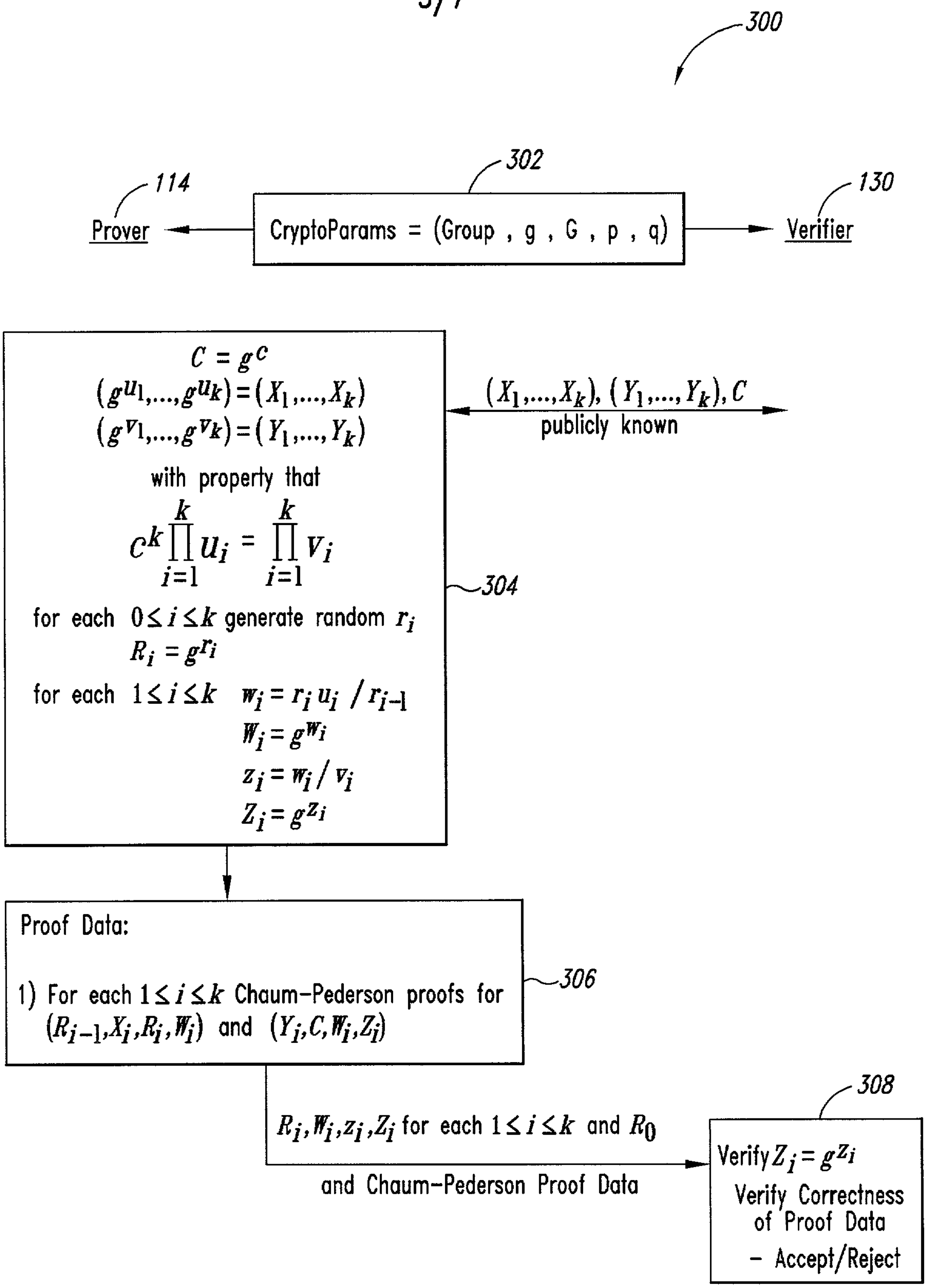

300
302
114
Prover
CryptoParams = (Group , g , G , p , q)
130
Verifier
$C = g^c$
$(g^{u_1},\ldots,g^{u_k}) = (X_1,\ldots,X_k)$
$(g^{v_1},\ldots,g^{v_k}) = (Y_1,\ldots,Y_k)$
with property that
$c^k \prod_{i=1}^{k} u_i = \prod_{i=1}^{k} v_i$
for each $0 \le i \le k$ generate random r_i
$R_i = g^{r_i}$
for each $1 \le i \le k$ $w_i = r_i u_i / r_{i-1}$
$W_i = g^{w_i}$
$z_i = w_i / v_i$
$Z_i = g^{z_i}$
304
$(X_1,\ldots,X_k), (Y_1,\ldots,Y_k), C$
publicly known
Proof Data:
1) For each $1 \le i \le k$ Chaum-Pederson proofs for (R_{i-1},X_i,R_i,W_i) and (Y_i,C,W_i,Z_i)
306
R_i,W_i,z_i,Z_i for each $1 \le i \le k$ and R_0
and Chaum-Pederson Proof Data
308
Verify $Z_i = g^{z_i}$
Verify Correctness of Proof Data
- Accept/Reject

Fig. 3

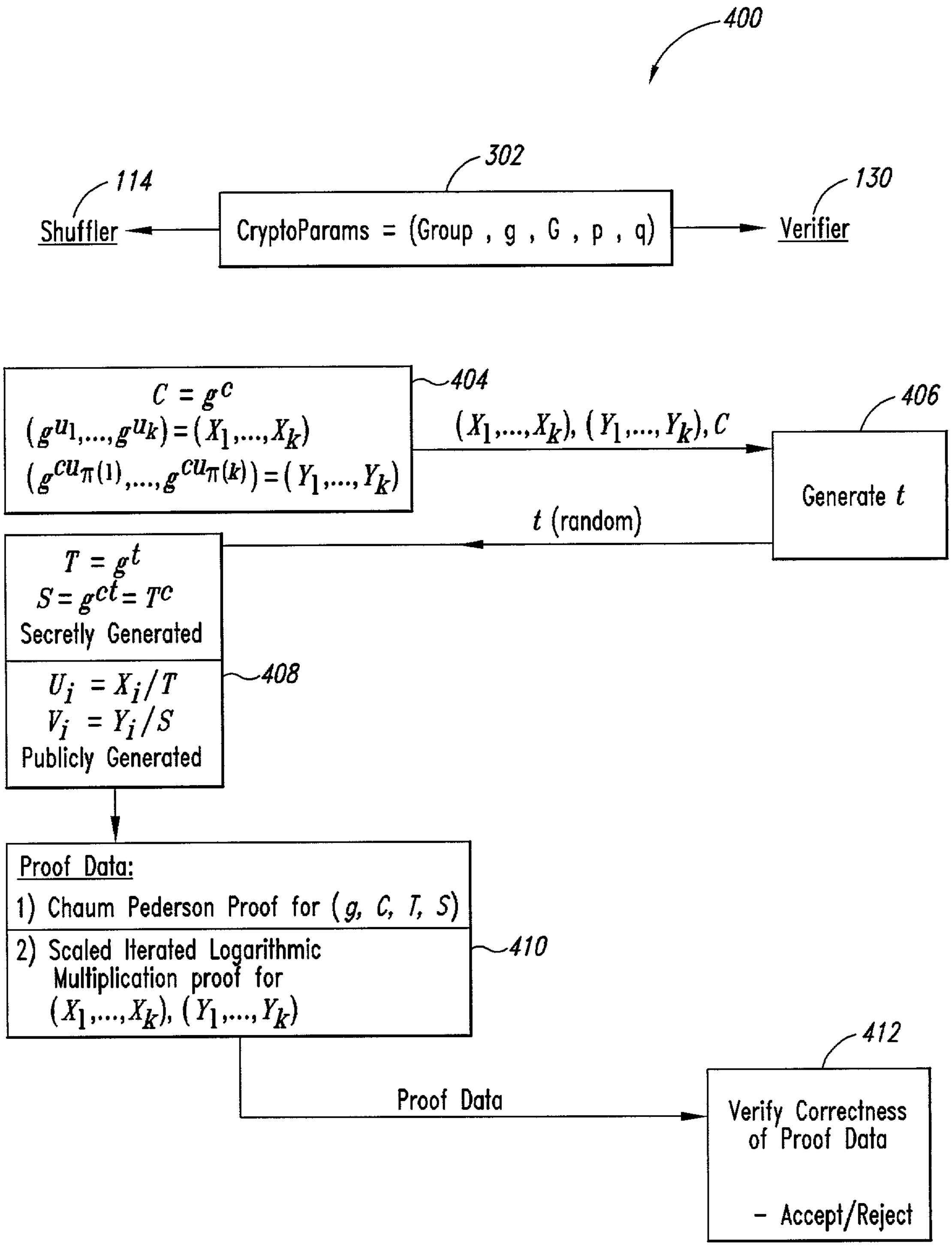

400
302
114
Shuffler
CryptoParams = (Group , g , G , p , q)
130
Verifier
$C = g^c$
$(g^{u_1},\dots,g^{u_k}) = (X_1,\dots,X_k)$
$(g^{cu_{\pi(1)}},\dots,g^{cu_{\pi(k)}}) = (Y_1,\dots,Y_k)$
404
$(X_1,\dots,X_k), (Y_1,\dots,Y_k), C$
406
Generate t
t (random)
$T = g^t$
$S = g^{ct} = T^c$
Secretly Generated
$U_i = X_i/T$
$V_i = Y_i/S$
Publicly Generated
408
Proof Data:
1) Chaum Pederson Proof for (g, C, T, S)
2) Scaled Iterated Logarithmic Multiplication proof for $(X_1,\dots,X_k), (Y_1,\dots,Y_k)$
410
Proof Data
412
Verify Correctness of Proof Data
– Accept/Reject

Fig. 4

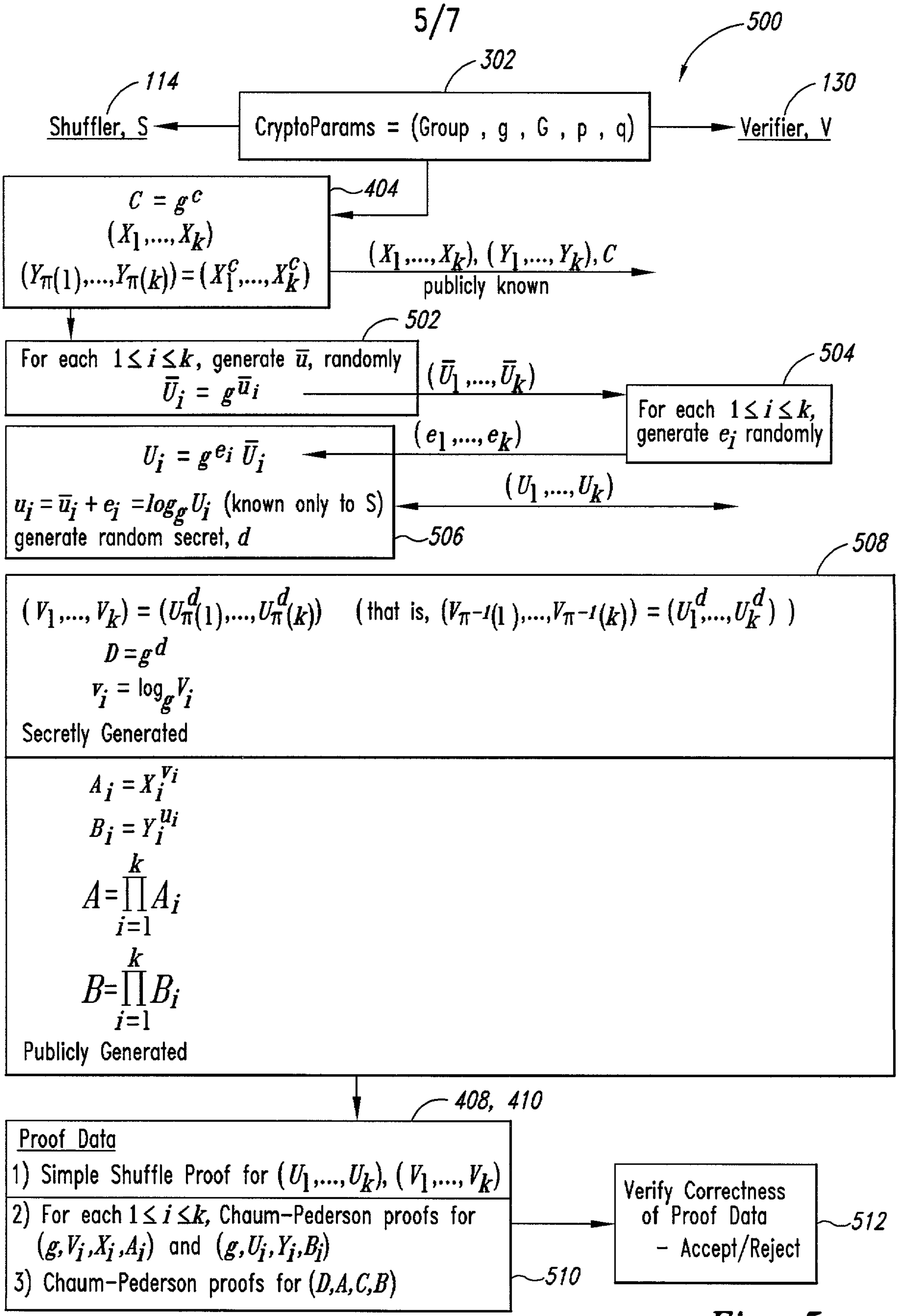

500
302
114
Shuffler, S
CryptoParams = (Group , g , G , p , q)
130
Verifier, V
404
$C = g^c$
$(X_1,\dots,X_k)$
$(Y_{\pi(1)},\dots,Y_{\pi(k)}) = (X_1^c,\dots,X_k^c)$
$(X_1,\dots,X_k)$, $(Y_1,\dots,Y_k)$, C
publicly known
502
For each $1 \le i \le k$, generate $\bar{u}$, randomly
$\bar{U}_i = g^{\bar{u}_i}$
$(\bar{U}_1,\dots,\bar{U}_k)$
504
For each $1 \le i \le k$, generate e_i randomly
$(e_1,\dots,e_k)$
$U_i = g^{e_i}\bar{U}_i$
$u_i = \bar{u}_i + e_i = \log_g U_i$ (known only to S)
generate random secret, d
$(U_1,\dots,U_k)$
506
508
$(V_1,\dots,V_k) = (U_{\pi(1)}^d,\dots,U_{\pi(k)}^d)$ (that is, $(V_{\pi^{-1}(1)},\dots,V_{\pi^{-1}(k)}) = (U_1^d,\dots,U_k^d)$)
$D = g^d$
$v_i = \log_g V_i$
Secretly Generated
$A_i = X_i^{v_i}$
$B_i = Y_i^{u_i}$
$A = \prod_{i=1}^{k} A_i$
$B = \prod_{i=1}^{k} B_i$
Publicly Generated
408, 410
Proof Data
1) Simple Shuffle Proof for $(U_1,\dots,U_k)$, $(V_1,\dots,V_k)$
2) For each $1 \le i \le k$, Chaum-Pederson proofs for (g, V_i, X_i, A_i) and (g, U_i, Y_i, B_i)
3) Chaum-Pederson proofs for (D,A,C,B)
510
Verify Correctness of Proof Data – Accept/Reject
512

Fig. 5

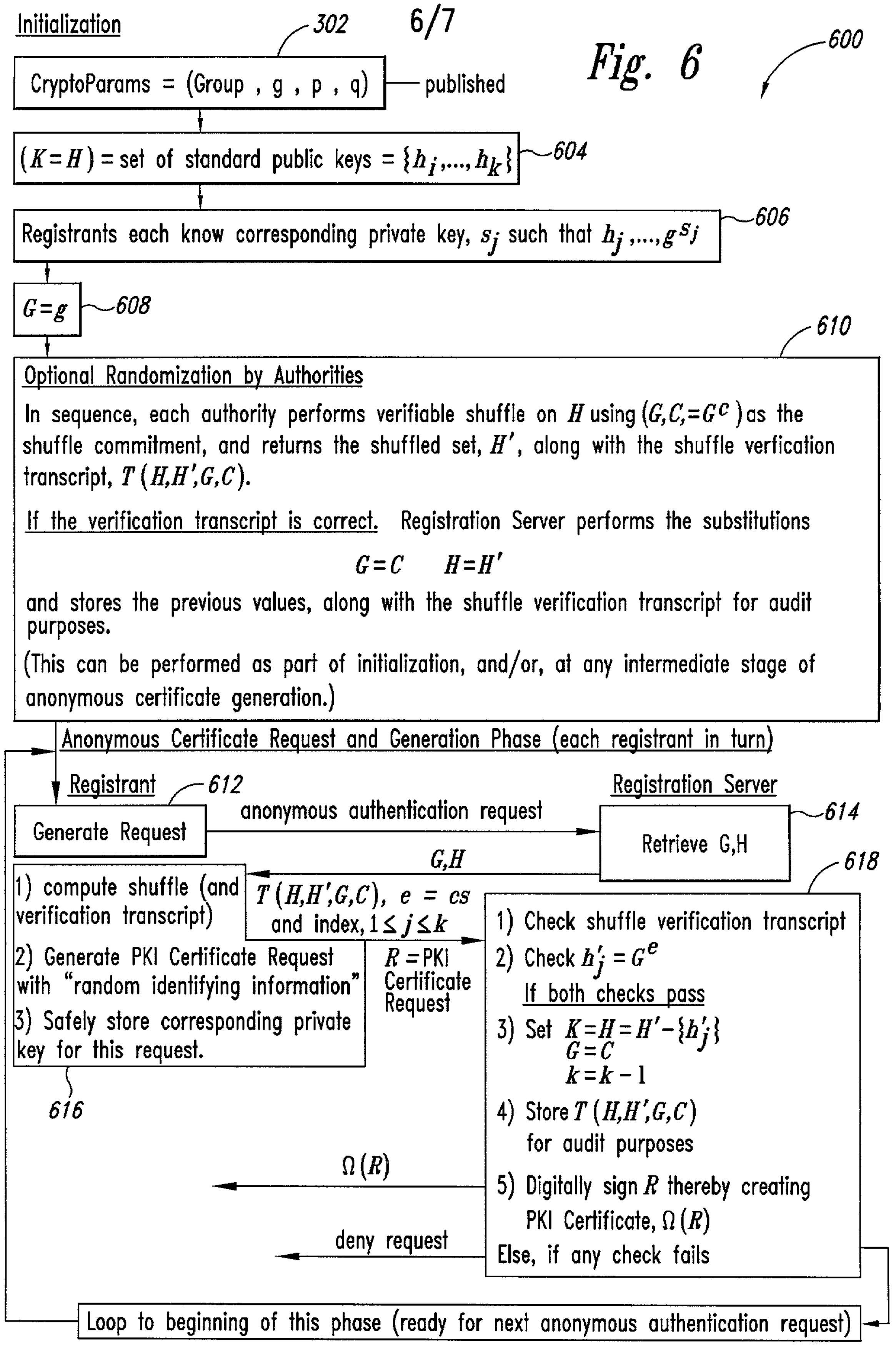
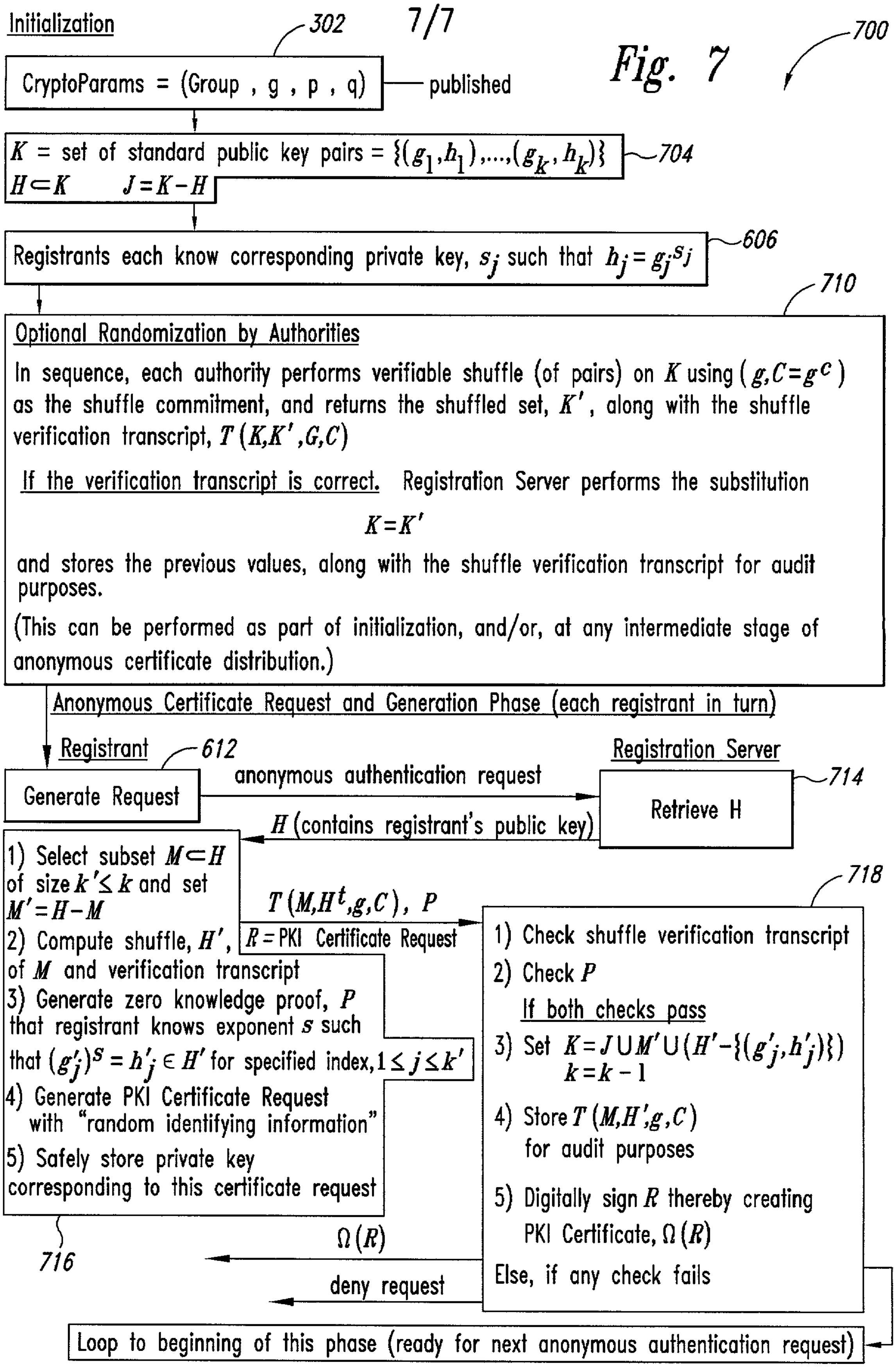

Initialization
302
CryptoParams = (Group , g , p , q)
published
600
(K=H) = set of standard public keys = {h_i,..., h_k}
604
Registrants each know corresponding private key, s_j such that h_j ,..., g^{s_j}
606
G=g
608
610
Optional Randomization by Authorities
In sequence, each authority performs verifiable shuffle on H using (G,C,=G^c) as the shuffle commitment, and returns the shuffled set, H′, along with the shuffle verfication transcript, T (H,H′,G,C).
If the verification transcript is correct. Registration Server performs the substitutions
G=C H=H′
and stores the previous values, along with the shuffle verification transcript for audit purposes.
(This can be performed as part of initialization, and/or, at any intermediate stage of anonymous certificate generation.)
Anonymous Certificate Request and Generation Phase (each registrant in turn)
Registrant
612
Registration Server
Generate Request
anonymous authentication request
Retrieve G,H
614
G,H
618
1) compute shuffle (and verification transcript)
2) Generate PKI Certificate Request with "random identifying information"
3) Safely store corresponding private key for this request.
616
T (H,H′,G,C), e = cs and index, 1 ≤ j ≤ k
R = PKI Certificate Request
1) Check shuffle verification transcript
2) Check h′_j = G^e
If both checks pass
3) Set K=H=H′−{h′_j}
G=C
k=k−1
4) Store T (H,H′,G,C) for audit purposes
5) Digitally sign R thereby creating PKI Certificate, Ω (R)
Else, if any check fails
Ω (R)
deny request
Loop to beginning of this phase (ready for next anonymous authentication request)

Fig. 6

Fig. 7


700
Initialization
302
CryptoParams = (Group , g , p , q)
published
K = set of standard public key pairs = $\{(g_1,h_1),\ldots,(g_k,h_k)\}$
$H \subset K$ $J = K - H$
704
Registrants each know corresponding private key, s_j such that $h_j = g_j^{s_j}$
606
710
Optional Randomization by Authorities
In sequence, each authority performs verifiable shuffle (of pairs) on K using $(g, C = g^c)$ as the shuffle commitment, and returns the shuffled set, K', along with the shuffle verification transcript, $T(K,K',G,C)$
If the verification transcript is correct. Registration Server performs the substitution
$K = K'$
and stores the previous values, along with the shuffle verification transcript for audit purposes.
(This can be performed as part of initialization, and/or, at any intermediate stage of anonymous certificate distribution.)
Anonymous Certificate Request and Generation Phase (each registrant in turn)
Registrant
612
Registration Server
Generate Request
anonymous authentication request
Retrieve H
714
H (contains registrant's public key)
1) Select subset $M \subset H$ of size $k' \leq k$ and set $M' = H - M$
2) Compute shuffle, H', of M and verification transcript
3) Generate zero knowledge proof, P that registrant knows exponent s such that $(g'_j)^s = h'_j \in H'$ for specified index, $1 \leq j \leq k'$
4) Generate PKI Certificate Request with "random identifying information"
5) Safely store private key corresponding to this certificate request
716
$T(M,H^t,g,C)$, P
R = PKI Certificate Request
718
1) Check shuffle verification transcript
2) Check P
If both checks pass
3) Set $K = J \cup M' \cup (H' - \{(g'_j, h'_j)\})$
$k = k - 1$
4) Store $T(M,H',g,C)$ for audit purposes
5) Digitally sign R thereby creating PKI Certificate, $\Omega(R)$
$\Omega(R)$
Else, if any check fails
deny request
Loop to beginning of this phase (ready for next anonymous authentication request)